

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-174817

(P2000-174817A)

(43) 公開日 平成12年6月23日 (2000.6.23)

(51) Int.Cl. ⁷	識別記号	F I	テマコード* (参考)
H 0 4 L 12/56		H 0 4 L 11/20	1 0 2 Z
H 0 4 Q 3/00		H 0 4 Q 3/00	
11/04		11/04	R

審査請求 有 請求項の数18 O L (全 16 頁)

(21) 出願番号	特願平11-172584	(71) 出願人	000004237 日本電気株式会社 東京都港区芝五丁目7番1号
(22) 出願日	平成11年6月18日 (1999.6.18)	(72) 発明者	ゴバラクリシナン ラママーシー アメリカ合衆国, ニュージャージー, 08540 プリンストン, インディペンデン ス ウエイ 4, エヌ・イー・シー・ユ ー・エス・エー・インク内
(31) 優先権主張番号	0 9 / 2 0 6 9 7 5	(74) 代理人	100097157 弁理士 桂木 雄二
(32) 優先日	平成10年12月8日 (1998.12.8)		
(33) 優先権主張国	米国 (U S)		

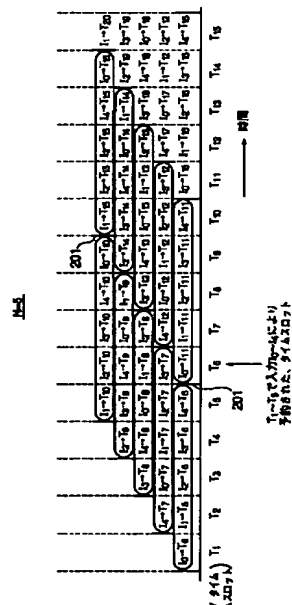
最終頁に続く

(54) 【発明の名称】 交換システムにおけるスケジューリング方法及び装置

(57) 【要約】

【課題】 超高速交換システムにおいて入出力間スケジュールを高速に確立することができる新規なスケジューリング方法及び装置を提供する。

【解決手段】 $N \times N$ 交換システムにおいて、 N 入力にそれぞれ対応した論理的キューと、 N 入力にそれぞれ対応し隣接アービタへ制御情報を送出する N 個のアービタとが設けられている。複数のスケジューリング過程がタイミングをずらせて開始され、 N タイムスロットだけ未来のタイムスロットがそれぞれ決定される。開始アービタからラウンドロビン方式で順次1つのアービタが選択され、選択されたアービタは未来のタイムスロットにおける利用可能な出力の集合の中から1つの出力を選択する。選択された出力を出力集合から除外し、択された入力とそれに関連する選択された出力との組をスケジュールとしてメモリに記憶する。複数のスケジューリング過程をパイプライン処理により同時に並列処理できるために、スケジュールを高速に確立することができる。



【特許請求の範囲】

【請求項1】 N入力及びN出力（Nは2以上の整数）を有し、各入力は前記N出力にそれぞれ対応したN個の論理的待ち行列（論理的キュー）からなる交換システムにおけるスケジューリング方法において、

a) 任意のスケジューリング過程を開始する入力に対して、当該開始入力から予め定められた数のタイムスロットだけ未来のタイムスロットを決定し、

b) 前記開始入力からラウンドロビン方式で順次1つの入力を選択し、

c) 選択された入力に転送すべきバケットが存在すれば、当該選択された入力に対して、前記未来のタイムスロットにおける利用可能な出力の集合の中から1つの出力を選択し、

d) 選択された出力を前記出力集合から除外し、前記選択された入力とそれに関連する前記選択された出力との組をスケジュールとして記憶する、ステップからなることを特徴とするスケジューリング方法。

【請求項2】 複数のスケジューリング過程がパイプライン処理により同時に進行することを特徴とする請求項1記載のスケジューリング方法。

【請求項3】 Nが奇数の場合、前記未来のタイムスロットは前記開始入力からNタイムスロットだけ未来に位置し、前記選択された出力を前記出力集合から除外した出力集合を次に選択されるべき入力へ転送する、ことを特徴とする請求項1記載のスケジューリング方法。

【請求項4】 N個のスケジューリング過程が1タイムスロットずつ位相をずらしながら同時に進行することを特徴とする請求項3記載のスケジューリング方法。

【請求項5】 Nが偶数の場合、前記未来のタイムスロットは前記開始入力から（N+1）タイムスロットだけ未来に位置し、前記選択された出力を前記出力集合から除外した出力集合を次に選択されるべき入力へ転送する動作を（N+1）タイムスロットのうち1回だけ1タイムスロット分遅延させる、ことを特徴とする請求項1記載のスケジューリング方法。

【請求項6】 （N+1）個のスケジューリング過程が1タイムスロットずつ位相をずらしながら同時に進行することを特徴とする請求項5記載のスケジューリング方法。

【請求項7】 前記交換システムは、更に、マルチキャスト・バケットを格納するための別個のキューを有し、各キューは当該キューのHOLバケットのあて先を示すマルチキャストビットマップ(BM)を有し、マルチキャスト・キューはユニキャスト・キューよりも優先されて処理され、任意の入力は前記出力集合とマルチキャストビットマップとの積集合を満たす全ての出力を選択し、

HOLマルチキャストバケットは、前記未来のタイムスロ

ットにおいて選択された全ての出力へ送信される、

ことを特徴とする請求項1ないし6のいずれかに記載のスケジューリング方法。

【請求項8】 N入力及びN出力（Nは2以上の整数）を有し、各入力は前記N出力にそれぞれ対応したN個の論理的待ち行列（論理的キュー）からなる交換システムにおいて、

前記N入力の各々は、

到着バケットを格納して前記論理的キューを管理する入力格納手段と、

前記到着バケットの出力要求に応じて、利用可能な出力の集合の中から1つの出力を選択して前記入力格納手段へ送出するアービタ手段と、

前記アービタ手段により選択された出力とその入力との組を接続情報として格納する接続格納手段と、

前記アービタ手段を隣接するアービタ手段と協調して制御するパイプライン制御手段と、

からなり、

前記パイプライン制御手段は、

10 a) 任意のスケジューリング過程を開始する入力のアービタ手段に対して、当該開始アービタ手段から予め定められた数のタイムスロットだけ未来のタイムスロットを決定し、

b) 前記開始アービタ手段からラウンドロビン方式で順次1つのアービタ手段を選択し、

c) 選択されたアービタ手段に対応する入力格納手段から出力要求があれば、当該選択されたアービタ手段により、前記未来のタイムスロットにおける利用可能な出力の集合の中から1つの出力を選択させ、

30 d) 選択された出力を前記出力集合から除外し、前記選択されたアービタ手段とそれに関連する前記選択された出力との組をスケジュールとして前記接続格納手段へ格納する、

ことを特徴とするスケジューリング装置。

【請求項9】 N入力及びN出力（Nは2以上の整数）を有するN×Nクロスバスイッチを有する交換システムにおいて、

前記N入力の各々は、

到着バケットを格納して前記N出力にそれぞれ対応したN個の論理的待ち行列（論理的キュー）管理する入力モジュールと、

前記到着バケットの出力要求に応じて、利用可能な出力の集合の中から1つの出力を選択して前記入力格納手段へ送出するアービタと、

前記アービタにより選択された出力とその入力との組を格納する接続メモリと、

前記アービタを隣接するアービタと協調して制御するパイプライン制御部と、

からなり、

40 前記N出力の各々は出力モジュールからなり、

前記パイプライン制御手段は、

a) 複数のスケジューリング過程を順次開始するアービタに対して、当該開始アービタから予め定められた数のタイムスロットだけ未来のタイムスロットをそれぞれ決定し、

b) 前記開始アービタからラウンドロビン方式で順次1つのアービタを選択し、

c) 選択されたアービタに対応する入力モジュールから出力要求があれば、当該選択されたアービタにより、前記未来のタイムスロットにおける利用可能な出力の集合の中から1つの出力を選択させ、

d) 選択された出力を前記出力集合から除外し、前記選択されたアービタとそれに関連する前記選択された出力との組をスケジュールとして前記接続メモリへ格納し、前記スケジュールに従って入力モジュールのバケットを前記クロスバースイッチを通して選択された出力へ送出することを特徴とする交換システム。

【請求項10】 N入力及びN出力(Nは2以上の整数)を有し、各入力は前記N出力にそれぞれ対応したN個の論理的待ち行列(論理的キュー)からなるラウンド・ロビン・グリーディ・スケジューリングプロトコルのためのクロスバースイッチにおけるタイムスロット決定方法であって、前記プロトコルの入力は全ての入出力キューの状態であり、前記プロトコルの出力はスケジュールであり、

a) $i = (\text{定数} - k - 1) \bmod N$ に対応する入力を選択し、

b) もし入力が無ければ停止し、それ以外であればラウンドロビンのやり方で $i = (i + 1) \bmod N$ により決定される次の入力を選択し、

c) 集合 $C = \{(i, j) | \text{出力 } j \text{ に対応する入力 } i \text{ において少なくとも1個のバケットが存在する}\}$ の要素である組 (i, j) が存在するならば、出力 j を選択し、

d) ステップc)において前記組 (i, j) が存在しなければ、入力集合から i を除去してステップb)に戻る、

e) 入力集合から i を、出力集合から j をそれぞれ除去し、

f) 前記組 (i, j) を前記スケジュールに加えてステップb)に戻る、

ステップからなることを特徴とするタイムスロットの決定方法。

【請求項11】 各タイムスロットにおいて、N個の異なるスケジュールが未来のNタイムスロット間で同時進行するスケジューリング方法において、

a) ラウンドロビン方式で、スケジューリングのための入力に対して特定の未来のタイムスロットを利用可能にし、

b) 入力 i によって、未来の k 番目のタイムスロットに対する出力を選択し、

c) 未来の k 番目のタイムスロットに対するスケジ

ュールを開始し、

d) 次の入力 $(i + 1) \bmod N$ を決定し、前記 k 番目のタイムスロットの間にバケットを自由に受信できる残りの出力を前記次の入力へ送出する、
ステップからなることを特徴とするスケジューリング方法。

【請求項12】 入力 i が k 番目のタイムスロットに対するスケジュールを完了しなかったならば、出力を選択し、更新された出力集合を次の入力へ送出し、前記入力 i が前記 k 番目のタイムスロットに対するスケジュールを完了するならば、前記出力集合を前記次の入力へ送出しない、ことを特徴とする請求項11記載の方法。

【請求項13】 出力の更新集合を前の入力から受信しない入力は、新たなスケジュールを開始することを特徴とする請求項12記載の方法。

【請求項14】 入力が奇数個の場合のパイプライン・ラウンドロビン・グリーディ・スケジューリング方法において、

e) $k(i, h) > 0$ は入力 i が h 番目のタイムスロットにおいて確保する出力のタイムスロットを示し、
 $i_h = (\text{定数} - N - h) \bmod N$ は h 番目のタイムスロットで新しくスケジューリングをはじめる入力を示し、
 $k(i, h) = 0$ は h 番目のタイムスロットでの入力 i の動作が抑制されることを意味するものとして、
 $k(0, 1) = k(1, 1) = \dots = k(N-1, 1) = 0$ 、定数 $= N+1$ に初期化し、

f) $O_h + N = \{0, 1, \dots, N-1\}$ 、 $0 \leq i \leq N-1$ 及び $i \neq i_h$ として、
 $k(i_h, h) = h + N$ 及び $k(i, h) = k((i-1) \bmod N, h-1)$ と設定し、

g) $0 \leq i \leq N-1$ である入力 i でバケットを送信すべき出力 j を集合 $O_{k(i, h)}$ からラウンドロビン方式で選択し(ただし $k(i, h) \neq 0$ である)、 j を $O_{k(i, h)}$ から除外し、

h) $0 \leq i \leq N-1$ である入力 i で選択された出力 j のアドレスをコネクションメモリのメモリ位置 $k(i, h) \bmod N$ に記憶し、対応する受信入出力キューからライン先頭のHOLバケットが別個の送信入出力キューへ移動し、

i) $0 \leq i \leq N-1$ で $i \neq (i_h - 2) \bmod N$ である入力 i で、次の入力 $(i + 1) \bmod N$ へ集合 $O_{k(i, h)}$ を転送し、

j) $0 \leq i \leq N-1$ である前記入力 i と、入力 i のメモリロケーション $(h \bmod (N+1))$ から読み込まれたアドレスの出力との間にクロスバコネクションを確立し、

k) $0 \leq i \leq N-1$ である各入力 i について、スケジューリングされた送信入出力キューの先頭に保持されたバケットをスイッチコアを通して送信する、
ステップからなるプロセスを用いて h 番目のタイムスロットを完了させる方法。

【請求項15】 入力が偶数個の場合のパイプライン・ラウンドロビン・グリーディ・スケジューリング方法において、

m) $k(i, h) > 0$ は入力 i が h 番目のタイムス

ットにおいて確保する出力のタイムスロットを示し、 $i_h = (\text{定数} - N - h) \bmod N$ は h 番目のタイムスロットで新しくスケジューリングをはじめる入力を示し、 $k(i, h) = 0$ は h 番目のタイムスロットでの入力 i の動作が抑制されることを意味するものとして、 $k(0, 1) = k(1, 1) = \dots = k(N-1, 1) = 0$ 、定数 $= N+1$ に初期化し、

n) $O_{k(i, h)} = \{0, 1, \dots, N-1\}$ 、 $0 \leq i \leq N-1$ である i で集合 $\{i_h, (i_h + 1) \bmod N\}$ の要素でないとして、 $k(i_h, h) = h + N + 1$ 、 $k((i_h + 1) \bmod N, h) = k(i_h, h - 2)$ 、及び $k(i, h) = k((i - 1) \bmod N, h - 1)$ に設定し、

o) $0 \leq i \leq N-1$ である入力 i においてパケットを送信すべき出力 j を集合 $O_{k(i, h)}$ からラウンドロビン方式で選択し（ただし、 $k(i, h) \neq 0$ である）、 j を $O_{k(i, h)}$ から除外し、

p) $0 \leq i \leq N-1$ である入力 i において、選択された出力 j のアドレスをコネクションメモリのメモリ位置 $k(i, h) \bmod (N+1)$ に記憶し、対応する受信出力キューからライン先頭の HOL パケットが別個の送信出力キューへ移動させ、

q) $0 \leq i \leq N-1$ で $i \neq (i_h - 2) \bmod N$ である入力 i において、 $(i_h - 2) \bmod N$ の入力が集合 $O_{k((i_h - 2) \bmod N, h)}$ を 1 タイムスロット分だけ遅延させた後、次の入力 $(i + 1) \bmod N \in$ 集合 $O_{k(i, h)}$ を転送し、

r) $0 \leq i \leq N-1$ である入力 i と、入力 i のメモリ位置 $(h \bmod (N+1))$ から読み込まれたアドレスの出力との間にクロスバーコネクションを確立し、

s) $0 \leq i \leq N-1$ である各入力 i について、スケジュールされた送信出力キューの先頭のパケットをスイッチコアを通して送信する、ステップからなるプロセスを用いて h 番目のタイムスロットを完了させる方法。

【請求項 16】 マルチキャスト・スケジューリングがラウンドロビン・グリーディ・スケジューリング・アルゴリズムに組み込まれており、マルチキャストパケットは到着順に処理される方式で格納され、ユニキャストキューよりも優先的に処理され、 h 番目のタイムスロットにおけるステップは、更に、

u) $0 \leq i \leq N-1$ である入力 i において、 $j \in O_{k(i, h)} \cap BM_i$ を満たす全ての出力 j を選択し、HOL マルチキャストパケットを k 番目のタイムスロットで選択された出力へ送信し、

v) $O_{k(i, h)} \cap BM_i$ が空集合であればユニキャストキューを処理し、それ以外の場合は、選択された出力を $O_{k(i, h)}$ 及び BM_i から除外し、

w) BM_i が空であれば、HOL マルチキャストパケットをマルチキャストキューから削除する、ステップからなることを特徴とする請求項 14 記載の方法。

【請求項 17】 マルチキャストスケジューリングがラウンドロビン・グリーディ・スケジューリング・アルゴリズムに組み込まれており、マルチキャストパケットは到着順に処理される方式で格納され、ユニキャストキューよりも優先的に処理され、 h 番目のタイムスロットにおけるステップは、更に、

x) $0 \leq i \leq N-1$ である入力 i において、 $j \in O_{k(i, h)} \cap BM_i$ を満たす全ての出力 j を選択し、HOL マルチキャストパケットを k 番目のタイムスロットで選択された出力へ送信し、

y) $O_{k(i, h)} \cap BM_i$ が空集合であればユニキャストキューを処理し、それ以外の場合は、選択された出力を $O_{k(i, h)}$ 及び BM_i から除外し、

z) BM_i が空であれば、HOL マルチキャストパケットをマルチキャストキューから削除する、ステップからなることを特徴とする請求項 15 記載の方法。

【請求項 18】 ステージ i が入力 I に関連し、前記ステージ i は未来のタイムスロットでの出力への送信をスケジューリングし、前記未来のタイムスロットは全てのステージを通して順次ずれて行く $N \times N$ スイッチをスケジューリングするための N ステージ・パイプラインシステムにおいて、

入力に対応する全てのパイプラインステージは、2つの入力が同時に同一の未来のタイムスロットを選択しないように、同時にスケジューリングを実行し、出力スロットはラウンドロビン方式に基づいて選択され、出力があるステージにより選択されるとき、当該出力は、パイプラインステージが入力によってあるタイムスロットで既に選択された出力を選択しないように、出力のフリーブールから除去される、ことを特徴とする N ステージ・パイプラインシステム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は電子及び光学メディア用のアプリケーションで使用する超高速（テラビット級）交換システムに係り、特に、超高速交換システムにおける入出力間のスケジューリング及びそのスケジューリングを実現するための交換システムに関する。

【0002】

【従来の技術】 広帯域化の要求に伴い、テラビットスイッチングの必要性が益々高まってきており、このような高速スイッチングに関する提案が多くなされている。ベシャイ及びミインタ（M. Beshai and E. Miinter）による “Multi-tera-bit/s switchbased on burst transfer and independent shared buffers,” ICC’ 95, pp.1724-1730、マッケOWN 等（N. McKeown et al）による

“The tiny tera: a packet switch core,” IEEE Micro, vol.171, Jan.-Feb. 1997, pp.26-33、及びゾン、シマズ、ツクダ及びユキマツ（W. D. Zhong, Y. Shimazu,

M.Tsukuda, and K. Yukimatsu) による "A modular Tbit/s TDM-WDM photonic ATM switch using optical buffers," IEICE Transactions on Communications, vol. E77-B, no.2, Feb. 1994, pp.190-196 を参照されたい。

【0003】電子的に制御される光スイッチングコア（見方を変えれば論理的クロスバスイッチ）は高性能スイッチの候補として魅力的である。10Gbpsのライン速度で、64バイトのセル／バケットが40ns以内に処理される必要があるからである。

【0004】当業者にとって重要な問題の1つは、このような光スイッチングコアを効率的に使用するスケジューリングの高速決定法の確立である。この場合、スイッチ（交換機）の設計において、入力バッファリング、出力バッファリングあるいはそれら両方を用いることが可能である。出力バッファリングを用いるスイッチでは、出力バッファのアクセス速度がスイッチ全体のスループットを上回っている必要がある。

【0005】このような出力バッファに必要とされる速度を抑えるために、ノックアウト法が採用される。この方法では、有限個のセルのみが出力バッファに受け入れられ、残りは廃棄される。光ノックアウトスイッチは、ゾン、シマズ、ツクダ及びユキマツ (Zhong, Y. Shimazu, M.Tsukuda, and K. Yukimatsu) による "A modular Tbit/s TDM-WDM photonic ATM switch using optical buffers," IEICE Transactions on Communications, vol. E77-B, no.2, Feb. 1994, pp.190-196で提案されている。光ノックアウトスイッチでは、各出力がいくつかの光逆Banyan網と光バッファを必要とするために構成が複雑となる。

【0006】入力バッファを用いるスイッチはもっと効率的にバッファを使用でき、メモリの帯域幅もライン速度の2倍で済む。簡単な方法では、各入力とその入力の待ち行列（キュー）における最初のバケットを送信する要求を出す。もし2以上の入力がある出力ポートを要求した場合、そのうちの1つがランダムに選ばれる。この方法は、カロール等 (M. J. Karol, M.G. Hiuchy, and S.P. Morgan) "Input vs. output queuing on a space-division packet switch," IEEE Transactions on Communications, vol. COM-35, no.12, Dec. 1987, pp. 1347-1356に示されており、この入力バッファリングアルゴリズムによって、均一トラフィック状況で0.587のスループットが得られている。トラフィックが均一でない場合は、スループットは更に低下する。

【0007】他のいくつかのスケジューリング法においては、HOL (Head Of Line: ラインの先頭) バケット以外のバケットが出力ポートを要求する。ファン、アキヤマ及びタナカ (R. Fan, M. Akiyama, and Y. Tanaka) による "An input buffer-type ATM switching using schedule comparison," Electronics and Communications

in Japan: Part I, vol.74, no.11, 1991, pp.17-25; モトヤマ、ベータ及びフロスト (S. Motoyama, D.W. Petr, and V.S. Frost) による "Input-queued switch based on a scheduling algorithm," Electronics Letters, vol.31, no.14, July 1995, pp.1127-1128; オバラ (H. Obara) による "Optimum architecture for input queuing ATM switches," Electronics Letters, vol.27, no.7, March 1991, pp.555-557を参照されたい。

【0008】より詳しく言えば、各タイムスロットにおいて、1つの入力に複数の出力ポートを要求する。タイムスロット毎にちょうど4つの要求を出せば、1に近い効率を得られる。しかしながら、この方法では、高速時、スケジューリングのための複数の要求及びその確認応答を1タイムスロット内で処理することができない（ここで1スロットはバケット送信時間を表す）。また、トラフィックに急激な変化がある場合には、各入力がある出力を要求するかを独立して決定することからスイッチパフォーマンスは低下するであろう。

【0009】スイッチパフォーマンスを改善したければ、スイッチコントローラがすべての入出力バッファのキューを知っていればよい。この情報によって、スイッチコントローラは各タイムスロットでの同時送信数を増加させることができる。しかしながら、SLIPプロトコルにおいては、複数の出力が独立して複数の入力に許可を与えるので、非効率的になっている。詳しくは、N. McKeown et al. "Scheduling cells in an input-queued switch," Electronic Letters, vol.29, no.25, Dec. 1993, pp.2174-2175を参照されたい。

【0010】また、アルゴリズムによって入力間のより良い協調を達成している例としては、D. Guo, Y. Yemini, Z. Zhang, "Scalable high-speed protocols for WDM optical star networks," IEEE INFOCOM 94を参照されたい。ただし、このアルゴリズムはスケジューリングを決定するのに多くのタイムスロットを必要とする点で不利である。

【0011】また、良好なパフォーマンスを示すランダム・グリーディ・スケジューリング法(RGS)が提案されている。これについては、R. Chipalkatti, Z. Xhang, and A. A. Acampora, "Protocols for optical star-coupler network using WDM: performance and complexity study," IEEE Journal on Selected Areas in Communications, vol.11, no.4, May 1993, pp.579-589、及びD. Guo, Y. Yemini, Z. Zhang, "Scalable high-speed protocols for WDM optical star networks," IEEE INFOCOM 94を参照されたい。

【0012】この従来のRGS法では、入力及びそれに対応するマッチングした出力の両方がランダムに選択される。しかしながら、実際には、このようなランダム選択を実装するのは困難である。各タイムスロットにおいて、N個のバケットがN個の入力からN個の出力へ転送さ

れ得ることに注意されたい。

【0013】

【発明が解決しようとする課題】上述したように、光スイッチングコアを効率的に使用するためのスケジューリング決定方法としては未だ満足すべきものがない。すなわち、入出力バッファリングを用いるものでは、構成の複雑化やスイッチのスループットの低下を引き起こす。また、スイッチコントローラが入出力バッファのキューを集中管理する方式では、スイッチの入力数及び出力数が増大するに伴って計算量が著しく増大し、1タイムスロット内で処理を完了することが困難となる。

【0014】そこで、本発明の目的は、超高速交換システムにおいて入出力間スケジュールを高速に確立することができる新規なスケジューリング方法及び装置（ラウンドロビン・グリーディ・スケジューリング：以下、RRGSという。）を提供することにある。

【0015】本発明の他の目的は、交換機の内部速度を上昇させることなく厳しいタイミング要求を満たすと共に良好なパフォーマンスを得ることができる新規なパイプラインアーキテクチャを有する交換システムを提供することにある。

【0016】

【課題を解決するための手段】本発明によるスケジューリング方法は、N入力及びN出力（Nは2以上の整数）を有し、各入力には前記N出力にそれぞれ対応したN個の論理的待ち行列（論理的キュー）からなる交換システムにおいて、a）任意のスケジューリング過程を開始する入力に対して、当該開始入力から予め定められた数のタイムスロットだけ未来のタイムスロットを決定し、b）前記開始入力からラウンドロビン方式で順次1つの入力を選択し、c）選択された入力に転送すべきバケットが存在すれば、当該選択された入力に対して、前記未来のタイムスロットにおける利用可能な出力の集合の中から1つの出力を選択し、d）選択された出力を前記出力集合から除外し、前記選択された入力とそれに関連する前記選択された出力との組をスケジュールとして記憶する、ステップからなることを特徴とする。更に、複数のスケジューリング過程がパイプライン処理により同時に進行することを特徴とする。

【0017】Nが奇数の場合には、前記未来のタイムスロットは前記開始入力からNタイムスロットだけ未来に位置し、前記選択された出力を前記出力集合から除外した出力集合を次に選択されるべき入力へ転送する。

【0018】Nが偶数の場合には、前記未来のタイムスロットは前記開始入力から（N+1）タイムスロットだけ未来に位置し、前記選択された出力を前記出力集合から除外した出力集合を次に選択されるべき入力へ転送する動作を（N+1）タイムスロットのうち1回だけ1タイムスロット分遅延させる。

【0019】本発明によるタイムスロット決定方法は、

N入力及びN出力（Nは2以上の整数）を有し、各入力には前記N出力にそれぞれ対応したN個の論理的待ち行列（論理的キュー）からなるラウンド・ロビン・グリーディ・スケジューリングプロトコルのためのクロスバースイッチにおけるタイムスロット決定方法であって、前記プロトコルの入力は全ての入出力キューの状態であり、前記プロトコルの出力はスケジュールであり、

a) $i = (\text{定数} - k - 1) \bmod N$ に対応する入力を選択し、

b) もし入力が無ければ停止し、それ以外であればラウンドロビンのやり方で $i = (i + 1) \bmod N$ により決定される次の入力を選択し、

c) 集合 $C = \{(i, j) \mid \text{出力} j \text{ に対応する入力} i \text{ において少なくとも1個のバケットが存在する}\}$ の要素である組 (i, j) が存在するならば、出力 j を選択し、

d) ステップc) において前記組 (i, j) が存在しなければ、入力集合から i を除去してステップb) に戻り、

e) 入力集合から i を、出力集合から j をそれぞれ除去し、

f) 前記組 (i, j) を前記スケジュールに加えてステップb) に戻る、

ステップからなることを特徴とする。

【0020】また、本発明によるスケジューリング方法は、各タイムスロットにおいて、N個の異なるスケジュールが未来のNタイムスロット間で同時進行するスケジューリング方法において、

a) ラウンドロビン方式で、スケジューリングのための入力に対して特定の未来のタイムスロットを利用可能にし、

b) 入力 i によって、未来の k 番目のタイムスロットに対する出力を選択し、

c) 未来の k 番目のタイムスロットに対するスケジュールを開始し、

d) 次の入力 $(i + 1) \bmod N$ を決定し、前記 k 番目のタイムスロットの間にバケットを自由に受信できる残りの出力を前記次の入力へ送出する、

ステップからなることを特徴とする。

【0021】本発明による入力が奇数個の場合のパイプライン・ラウンドロビン・グリーディスケジューリング方法は、

e) $k(i, h) > 0$ は入力 i が h 番目のタイムスロットにおいて確保する出力のタイムスロットを示し、 $i_h = (\text{定数} - N - h) \bmod N$ は h 番目のタイムスロットで新しくスケジューリングをはじめる入力を示し、 $k(i, h) = 0$ は h 番目のタイムスロットでの入力 i の動作が抑制されることを意味するものとして、 $k(0, 1) = k(1, 1) = \dots = k(N-1, 1) = 0$ 、定数 $= N+1$ に初期化し、

f) $O_h + N = \{0, 1, \dots, N-1\}$ 、 $0 \leq i \leq N-1$ 及び $i \neq i_h$ として、 $k(i_h, h) = h + N$ 及び $k(i, h) = k((i-1) \bmod N, h-1)$ と設定し、

g) $0 \leq i \leq N-1$ である入力 i でパケットを送信すべき出力 j を集合 $O_{k(i,h)}$ からラウンドロビン方式で選択し(ただし $k(i,h) \neq 0$ である)、 j を $O_{k(i,h)}$ から除外し、

h) $0 \leq i \leq N-1$ である入力 i で選択された出力 j のアドレスをコネクションメモリのメモリ位置 $k(i,h) \bmod N$ に記憶し、対応する受信入出力キューからライン先頭のHOLパケットが別個の送信入出力キューへ移動し、

i) $0 \leq i \leq N-1$ で $i \neq (i_h - 2) \bmod N$ である入力 i で、次の入力 $(i+1) \bmod N \in$ 集合 $O_{k(i,h)}$ を転送し、

j) $0 \leq i \leq N-1$ である前記入力 i と、入力 i のメモリロケーション $(h \bmod (N+1))$ から読み込まれたアドレスの出力との間にクロスバーコネクションを確立し、

k) $0 \leq i \leq N-1$ である各入力 i について、スケジューリングされた送信入出力キューの先頭に保持されたパケットをスイッチコアを通して送信する、ステップからなるプロセスを用いて h 番目のタイムスロットを完了させることを特徴とする。

【0022】更に、本発明による入力が偶数個の場合のバイブライン・ラウンドロビン・グリーディスケジューリング方法は、

m) $k(i, h) > 0$ は入力 i が h 番目のタイムスロットにおいて確保する出力のタイムスロットを示し、 $i_h = (\text{定数} - N - h) \bmod N$ は h 番目のタイムスロットで新しくスケジューリングをはじめる入力を示し、 $k(i, h) = 0$ は h 番目のタイムスロットでの入力 i の動作が抑制されることを意味するものとして、 $k(0, 1) = k(1, 1) = \dots = k(N-1, 1) = 0$ 、定数 $= N+1$ に初期化し、

n) $O_{i_h + N + 1} = \{0, 1, \dots, N-1\}$ 、 $0 \leq i \leq N-1$ である i で集合 $\{i_h, (i_h + 1) \bmod N\}$ の要素でないとして、 $k(i_h, h) = h + N + 1$ 、 $k((i_h + 1) \bmod N, h) = k(i_h, h - 2)$ 、及び $k(i, h) = k((i-1) \bmod N, h-1)$ に設定し、

o) $0 \leq i \leq N-1$ である入力 i においてパケットを送信すべき出力 j を集合 $O_{k(i,h)}$ からラウンドロビン方式で選択し(ただし、 $k(i,h) \neq 0$ である)、 j を $O_{k(i,h)}$ から除外し、

p) $0 \leq i \leq N-1$ である入力 i において、選択された出力 j のアドレスをコネクションメモリのメモリ位置 $k(i, h) \bmod (N+1)$ に記憶し、対応する受信入出力キューからライン先頭のHOLパケットが別個の送信入出力キューへ移動させ、

q) $0 \leq i \leq N-1$ で $i \neq (i_h - 2) \bmod N$ である入力 i において、 $(i_h - 2) \bmod N$ の入力が集合 $O_{k(i_h - 2) \bmod N, h}$ を1タイムスロット分だけ遅延させた後、次の入力 $(i+1) \bmod N \in$ 集合 $O_{k(i,h)}$ を転送し、

r) $0 \leq i \leq N-1$ である入力 i と、入力 i のメモリ位置 $(h \bmod (N+1))$ から読み込まれたアドレスの出力との間にクロスバーコネクションを確立し、

s) $0 \leq i \leq N-1$ である各入力 i について、スケジューリングされた送信入出力キューの先頭のパケットをスイッチコアを通して送信する、

ステップからなるプロセスを用いて h 番目のタイムスロットを完了させることを特徴とする。

【0023】更に、マルチキャストスケジューリングがラウンドロビン・グリーディ・スケジューリング・アルゴリズムに組み込まれており、マルチキャストパケットは到着順に処理される方式で格納され、ユニキャストキューよりも優先的に処理され、 h 番目のタイムスロットにおけるステップは、更に、

u) $0 \leq i \leq N-1$ である入力 i において、 $j \in O_{k(i,h)}$ 、 $\cap BM_i$ を満たす全ての出力 j を選択し、HOLマルチキャストパケットを k 番目のタイムスロットで選択された出力へ送信し、

v) $O_{k(i,h)}$ 、 $\cap BM_i$ が空集合であればユニキャストキューを処理し、それ以外の場合は、選択された出力を $O_{k(i,h)}$ 及び BM_i から除外し、

w) BM_i が空であれば、HOLマルチキャストパケットをマルチキャストキューから削除する、ステップからなることを特徴とする。

【0024】本発明による N ステージ・バイブラインシステムは、ステージ i が入力 I に関連し、前記ステージ i は未来のタイムスロットでの出力への送信をスケジューリングし、前記未来のタイムスロットは全てのステージを通して順次ずれて行く $N \times N$ スイッチをスケジューリングするための N ステージ・バイブラインシステムであって、入力に対応する全てのバイブラインステージは、2つの入力が同時に同一の未来のタイムスロットを選択しないように、同時にスケジューリングを実行し、出力スロットはラウンドロビン方式に基づいて選択され、出力があるステージにより選択されるとき、当該出力は、バイブラインステージが入力によってあるタイムスロットで既に選択された出力を選択しないように、出力のフリーブルから除去されることを特徴とする。

【0025】本発明によれば、 N 入力からラウンドロビン方式で順次1つの入力を選択し、その選択された入力に対して未来のタイムスロットにおける利用可能な出力の集合の中から1つの出力を選択し、スケジューリングとして記憶する。これにより、複数の出力割当動作(スケジューリング)を並列処理することが可能となり、内部速度の向上を必要とせず厳しいタイミング基準を満たすことができる。更に、並列処理により、RGCSの良好なパフォーマンスを損なわず、100%に近い使用率を達成することができる。

【0026】

【発明の実施の形態】図1は本発明による超高速交換システムの概念的アーキテクチャを示すブロック図である。本交換システムは $N \times N$ クロスバースイッチ101からなり、 N 本の入力ラインから受信するパケットは全て

固定長のセルである。N個の入力ポートはそれぞれN個のアービタを有し、更に各入力ポートはN個の出力ポートにそれぞれ対応したN個の論理キューを有する。後述するように、N個のアービタ及び各アービタに対応するN個の論理キューはRRGSアルゴリズムにより制御される。以下、RRGSスケジューリングについて詳細に説明する。

【0027】ラウンドロビン・グリーディ・スケジューリング (RRGS)

RRGSプロトコルの入力は、全ての入出力キューの状態である。クロスバスイッチ101の各入力ポートを i ($i \in \{0, 1, \dots, N-1\}$) と記すと、このような入力、即ち入出力キュー、は次のような集合Cによって表すことができる。

【0028】 $C = \{(i, j) \mid \text{出力}j\text{に対応する入力}i\text{において少なくとも1個のバケットが存在する}\}.$

【0029】RRGSプロトコルの出力は、N個の入力をN個の出力に対応づけるスケジュールである。このようなスケジュールの集合Sは次のように表すことができる。

【0030】 $S = \{(i, j) \mid \text{入力}i\text{から出力}j\text{へバケットが送られる}\}.$

【0031】当業者にとっては明らかであるが、各タイムスロットにおいて、1つの入力は1個のバケットのみを送信でき、1つの出力は1個のバケットのみを受信できる。この条件下で、任意のk番目のタイムスロットのスケジュールは、次のステップ1)～4)によって決定される。

【0032】ステップ1) 全入力の集合を $I_k = \{0, 1, \dots, N-1\}$ 、全出力の集合を $O_k = \{0, 1, \dots, N-1\}$ とする。 $i = (\text{定数} - k - 1) \bmod N$ の計算式によって、即ち (定数 - k - 1) をNで除算した剰余として入力iを選択する。スケジュールを開始する入力をこのように選択することでスケジューリングのインプリメンテーションを簡単にできるであろう。

【0033】ステップ2) もし I_k が空であれば、停止する。空でなければ、ラウンドロビンのやり方で $i = (i + 1) \bmod N$ となる次の入力iを選択する。

【0034】ステップ3) $(i, j) \in C_k$ であるように O_k から出力jをラウンドロビンのやり方で選択する。もしそのような出力が存在しなければ、 I_k からiを除去し、ステップ2)に戻る。

【0035】ステップ4) I_k から入力iを、 O_k から出力jを除去し、 S_k に (i, j) を加え、ステップ2)に戻る。

【0036】上記プロトコルは、明らかに上述した従来のRGS法の改良である。上述したように、従来のRGS法では、入力及びそれに対応するマッチングした出力の両方がランダムに選択されていた。しかし、実際には、このようなランダム選択を実装するのは難しい。各タイムスロットにおいて、N個のバケットがN個の入力からN個の

出力へ転送され得ることに注意されたい。

【0037】これに対して、RRSGでは、ある与えられたタイムスロットのスケジューリングプロセスはN個の位相からなる。あるタイムスロットのスケジューリングの各位相において、1つの入力は当該タイムスロットでの送信のために使用可能な残存する出力の1つを選択する。後述するように(図4参照)、1つの位相は、入力モジュール(IM)からラウンドロビン(RR)アービタへ出線リクエスト(要求)を送出し、RRアービタにおいてラウンドロビン方式で出力選択を行い、そしてRRアービタから入力モジュールIMへ要求応答(出線選択応答)を送出する、というステップからなる。入力が出力を選択するラウンドロビン(RR)順序は、すべての入力に対して等しいアクセスを保証するようにタイムスロット毎に循環的にシフトする。

【0038】奇数個入力のパイプラインRRGS

10Gbpsのような高速リンク速度においては、N位相を1つのタイムスロット(64バイトのバケットサイズを仮定すると40ns)内で終了させることは出来ない。リンク速度が高速化するにつれて、従来の技術では1タイムスロットで1位相を超える処理を完了することができない。

【0039】この問題を解決するため、本発明ではパイプライン方式を用いる。すなわち、タイムスロット毎に、未来のN個のタイムスロットにわたって、N個の異なるスケジュールが同時進行する。ある1つのスケジュールの各位相は1個の入力だけに関連する。言い換えれば、任意のタイムスロットにおいて、他の複数の入力は、他の異なる未来のタイムスロットに対するスケジュールの位相を実行することとなる。

【0040】定義：ある未来のタイムスロット T_k のスケジュールが完了したとは、N位相全てが完了したとき、すなわち、タイムスロット T_k において送信するための出力を選択するチャンス(成功/失敗に関わらず)がすべての入力に与えられたとき、をいう。

【0041】あるスケジュールのN個の位相を完了するのにN個のタイムスロットが必要であるが、後述するようにパイプラインアプローチを用いてN個のスケジュールを並列計算することにより、N個の異なるスケジュールのN個の位相を1タイムスロット内で完了させることができる。しかし、このことはタイムスロット毎に1つのスケジュールを完了させることと結果的に等価である。

【0042】言い換えれば、RRGSにおいては、ある未来のタイムスロットが全ての入力に対するラウンドロビンスケジューリングに利用可能となる。すなわち、未来のk番目のタイムスロット T_k のスケジュールを開始する入力iはラウンドロビン方式で出力を選択し、 $(i + 1) \bmod N$ である次の入力iに対して、当該k番目のタイムスロット T_k でバケットを自由に受信可能な出力ポートを示す集合 O_k を送出的る。

【0043】前の入力 $(i-1) \bmod N$ から k 番目のタイムスロット T_k でまだ利用可能な出力ポートの集合 Q_k を受け取った任意の入力 i は、もし可能であれば、この中から1つの出力を選択し、もし k 番目のタイムスロット T_k のスケジューリングが完了しなければ、更新された出力集合 Q_k を次の入力 $i = (i+1) \bmod N$ へ送出する。 k 番目のタイムスロット T_k のスケジューリングが完了したならば、更新された集合 Q_k は次の入力 $(i+1) \bmod N$ へ送出されない。これにより、現タイムスロットにおいて出力集合 Q_k を受け取らなかった入力 $(i+1) \bmod N$ は、次のタイムスロットで新たな未来のタイムスロットのための新しいスケジューリングを開始するであろう。

【0044】RRGSの上記ステップ(1)は、 N 個のタイムスロットの期間に1度、入力が出力集合 Q_k を送出しないことを含意している。集合 Q_k を送出しない入力 i は k 番目のタイムスロットにおける出力を選択する最後の入力である。

【0045】定理1 入力数 N が奇数で、 $(\text{定数}-k) \bmod N$ の入力が $k-1$ 番目のタイムスロットにおいて集合 Q_k を送出しないならば、 $(\text{定数}-k) \bmod N$ の入力は k 番目のタイムスロットのスケジューリングを完了させる。

【0046】(証明) 上記定理1は次のことを含意する。

① 各タイムスロットにおいて、 N 個すべての入力は、未来のあるタイムスロットでの送信をスケジューリングする機会を有する。

② 各タイムスロットにおいて、入力は未来の1を超えないタイムスロットでの送信をスケジューリングすることができる。

③ 各タイムスロットにおいて、ある出力は1個の入力のみから送信を受け付けるようにスケジュールされる。

【0047】 $(\text{定数}-k) \bmod N$ の入力 i を $k-1$ 番目のタイムスロットにおいて集合 Q_k を送出しない入力と仮定すると、それ以前の $N-1$ 個の入力の各々は k 番目のタイムスロットの予約をするときに集合 Q_k を送出しなければならない。なお、 $(k-1-j)$ 番目のタイムスロットにおいて、 $(i+j) \bmod N$ の入力は集合 Q_k を次の入力へ送出しない。また、 $(i-j) \bmod N$ の入力は k 番目のタイムスロットのための予約を行う。このようなスケジューリングは、 $1 \leq j \leq (N-1)$ の任意の j について、

$$\forall (1 \leq j \leq (N-1)) \quad i-j \neq i+j \bmod N \Leftrightarrow$$

$$\forall (1 \leq j \leq (N-1)) \quad 2j \neq 0 \bmod N \Leftrightarrow$$

N が奇数

であるならば、実現可能である。

【0048】従って、 k 番目のタイムスロットのスケジューリングが $k-N$ 番目のタイムスロットでの $(i+1) \bmod N$ の入力によって開始されたということは、 $k-N-1$ 番目のタイムスロットでの入力 i が集合 Q_{k-N} を送出なかったことを意味する。

(証明終)。

【0049】図2は、奇数入力数の 5×5 クロスバースイッチを用いた場合のRRGSスケジューリングの一例を示すタイミングチャートである。図2では、5つの入力 $I_0 \sim I_4$ と、それら入力が出力ポートを選択するタイムスロット T_1, T_2, \dots との関係が示されている。

【0050】図2において、例えばタイムスロット T_1 で、入力 I_1 はタイムスロット T_1 で送信を行うための出力ポートの選択(スケジューリング)を行い、入力 I_3 はタイムスロット T_3 におけるスケジューリングを行っている。また、次のタイムスロット T_2 では、入力 I_1 はタイムスロット T_2 におけるスケジューリングを行っている。以下同様である。図中に参照番号201で例示された太い縦線は、その前の入力でスケジューリングが完了し、次の入力で新しいスケジューリングが開始することを示している。関連するタイムスロットにおいて出力を選択する最後の入力である場合には、当該入力は次の入力へ集合 Q を送出しない。この条件は $N=5$ タイムスロット毎に発生するから、入力はモジュロ N カウンタによって集合 Q を送出しないことを決定することができる。

【0051】最後に、 h 番目のタイムスロット(例えば現在のタイムスロット)におけるRRGSの動作を説明する。 Q_k は k 番目のタイムスロットの利用可能な出力の集合を示す。 $k(i, h) > 0$ は、入力 i が h 番目のタイムスロットにおいて確保する出力のタイムスロットを示し、 $i_h = (\text{定数}-N-h) \bmod N$ は h 番目のタイムスロットで新しくスケジューリングをはじめる入力を示す。また、 $k(i, h) = 0$ は、 h 番目のタイムスロットでの入力 i の動作が抑制されることを意味する。スケジューラは適切な初期化が必要で、初期化期間は N タイムスロット続く。初期化が最初のタイムスロット T_1 で始まると仮定すると、初期化は $k(0,1) = k(1,1) = \dots = k(N-1,1) = 0$ 、 $\text{定数}=N+1$ と設定することにより開始される。つまり、それ以後更新されるまで最初の N タイムスロットはすべての入力の動作が抑制される。また、バケットは、入力ポートにおいて、論理的に分離したキュー(入出力キュー)で待機すると仮定する。この入出力キューでは各出力ポートに対応して1つのキューが設けられるために、HOL(Head Of Line: ライン先頭)ブロッキングが防止される。更に、受信入出力キュー及び送信入出力キューも設けられている。

【0052】図1に示すシステムに即して説明すれば、つぎのように定式化できる。

【0053】1) $Q_{h+N} = \{0, 1, \dots, N-1\}$ 、 $0 \leq i \leq N-1$ 及び $i \neq i_h$ として、 $k(i_h, h) = h + N$ 及び

$$k(i, h) = k((i-1) \bmod N, h-1)。$$

【0054】2) $0 \leq i \leq N-1$ である入力 i は、バケットを送信すべき出力 j を出力集合 $Q_{k(i,h)}$ からRR方式で選択する。ただし、 $k(i,h) \neq 0$ である。なお、 $0 \leq i \leq$

$N-1$ 及び $i \neq i_h$ である入力 i は先行するタイムスロットにおいて $(i-1) \bmod N$ の入力から集合 $O_{k(i,h)}$ を受信している。選択された出力 j は、受信した集合 $O_{k(i,h)}$ から除外される。

【0055】図1に示すシステムにおいて、 N 個のアービタ $(0, 1, \dots, N-1)$ の各々は、タイムスロット毎に順次前のアービタから、利用可能な出力ポートの情報、即ち集合 $O_{k(i,h)}$ を受け取り、その利用可能な出力ポートの中から1つの出力ポートをRR方式で選択する。選択された出力ポートは集合 $O_{k(i,h)}$ から除外される。こうして更新された集合 $O_{k(i,h)}$ は、後述するように次のアービタへ転送される。

【0056】3) $0 \leq i \leq N-1$ である入力 i は、選択された出力 j のアドレスをコネクションメモリのメモリ位置 $k(i, h) \bmod N$ に記憶する。対応する受信入出力キューからライン先頭のHOLバケットが別個の送信入出力キューへ移される。

【0057】4) $0 \leq i \leq N-1$ で $i \neq (i_h - 2) \bmod N$ である入力 i (現アービタ) は、次の入力 $(i+1) \bmod N$ (後続するアービタ) へ集合 $O_{k(i,h)}$ を転送する。ここでは、出力ポートの使用/不使用状況を示す N bit の情報だけを転送すればよい。

【0058】5) 続いて、クロスバースイッチ 101 は、 $0 \leq i \leq N-1$ である入力 i と、入力 i のメモリロケーション $(h \bmod (N+1))$ から読み込まれたアドレスの出力ポートとの間にクロスバコネクションを確立する。

【0059】6) $0 \leq i \leq N-1$ である各入力 i について、スケジューリングされた送信入出力キューの先頭 (HOL) バケットをスイッチコアに確立されたコネクションを通して選択された出力ラインへ送信する。

【0060】偶数個入力のバイブラインRRGS
上述したように、入力ポート数が奇数個の場合、各入力 i は、当該入力 i が未来のタイムスロットのための出力ポートを選択する最後のものであるならば、更新された集合 O_k を隣接入力へ転送せず、これにより当該タイムスロットのスケジューリングを完了する。しかしながら、この奇数個用のアルゴリズムを偶数の場合に直接適用すると、いくつかの入力は複数の未来のタイムスロットに対してスケジューリングを行い、逆に、他の入力は全くスケジューリングを実行しなくなってしまう。そのため、入力ポート数が偶数個のスイッチを制御するには、上述したバイブライン技法を修正する。

【0061】上述した定理1の証明から、制御情報をブロックする代わりに遅延させることで、入力が偶数個の場合に対応できると推論できる。入力が偶数個の場合は、各入力は N 個のタイムスロットで一回だけ集合 O_k を次へ送らず、次のタイムスロットにおいて、当該入力は前のタイムスロットから遅れた集合 O_k を送出する。入力 i が遅延した集合 O_k を転送するとき、現在の集合 O_k は転送しない。従って、当該入力 i は k 番目のタイムスロ

ットにおいて出力を選択する最後の入力となる。

【0062】定理2 入力数 N が偶数で、 $(\text{定数} - k) \bmod N$ の入力が $k-2$ 番目のタイムスロットにおける集合 O_k を遅延させ、 $k-1$ 番目のタイムスロットにおいて転送するならば、 $(\text{定数} - k) \bmod N$ の入力は k 番目のタイムスロットのスケジュールを完了させる。

【0063】(証明) $(\text{定数} - k) \bmod N$ の入力 i が $k-2$ 番目のタイムスロットにおける集合 O_k を遅延させ、 $k-1$ 番目のタイムスロットにおいて現集合 O_k の代わりに遅延された集合 O_k を転送するものと仮定する。この場合、 $(k-1-j)$ 番目のタイムスロットにおいて、 $(i+j-1) \bmod N$ の入力は集合 O_k を遅延させ、 $(i+j) \bmod N$ の入力は遅延された集合 O_k を次の入力へと転送する。

$(k-1-j)$ 番目のタイムスロットにおいて、 $(i-j) \bmod N$ の入力は k 番目のタイムスロットを予約し、集合 O_k を転送する。ただし、 N が偶数で、 $0 \leq j \leq N/2 - 1$ とした場合、 $i-j \neq i+j \bmod N$ 及び $i-j \neq i+j-1 \bmod N$ である。

【0064】 $(i-N/2) \bmod N$ の入力は、 k 番目のタイムスロットをどの入力も予約しないように、 $(k-1-N/2)$ 番目のタイムスロットにおいて集合 O_k を格納する。 $(k-2-N/2)$ 番目のタイムスロットにおいて、 $(i-N/2) \bmod N$ の入力は、 k 番目のタイムスロットを予約する。 $N/2 + 2 \leq j \leq N$ として、 $(k-1-N/2)$ 番目のタイムスロットにおいて、 $(i-j+1) \bmod N$ の入力は k 番目のタイムスロットを予約し、集合 O_k を転送する。ただし、 N が偶数で、 $i-j+1 \neq i+j \bmod N$ 及び $i-j+1 \neq i+j-1 \bmod N$ である。

【0065】従って、 k 番目のタイムスロットのスケジューリングは、入力 i に対応するユーザ i によって終了されるまで中断することなくバイブラインによって進行する。このスケジュールが $(k-N-1)$ 番目のタイムスロットで $(i+1) \bmod N$ のユーザ (入力) によって開始されたということは、当該入力が $(k-N-2)$ 番目のタイムスロットにおいて制御情報 (即ち、集合 O) の転送を遅延させたからである。

(証明終)。

【0066】図3は、偶数入力数の 4×4 クロスバースイッチを用いた場合のRRGSスケジューリングの一例を示すタイミングチャートである。図3では、4つの入力 $1, \dots, 4$ と、それら入力が出力ポートを選択するタイムスロット T_1, T_2, \dots との関係が示されている。

【0067】図中の太い縦線 301 は、その前の入力でスケジューリングが完了し次の入力から新しいスケジューリングを開始することを示し、斜線部 302 は制御情報 O が遅延していることを示す。

【0068】上述したように、 h 番目のタイムスロット (例えば現在のタイムスロット) におけるRRGSの動作を説明する。 O_k は k 番目のタイムスロットの利用可能な出

力の集合を示す。 $k(i, h) > 0$ は、入力 i が h 番目のタイムスロットにおいて確保する出力のタイムスロットを示し、 $i_h = (\text{定数} - N - 1 - h) \bmod N$ は h 番目のタイムスロットで新しくスケジューリングを開始する入力を示す。また、 $k(i, h) = 0$ は、 h 番目のタイムスロットでの入力 i の動作が抑制されることを意味する。スケジューラは適切な初期化が必要で、初期化期間は N タイムスロット続く。初期化が最初のタイムスロット T_1 で始めると仮定すると、初期化は $k(0, 1) = k(1, 1) = \dots = k(N-1, 1) = 0$ 、定数 $= N+2$ と設定することにより開始される。つまり、それ以後更新されない限り最初の N タイムスロットはすべての入力の動作が抑制される。

【0069】図1に示すシステムに即して説明すれば、つぎのように定式化できる。

【0070】1) $O_{h+N+1} = \{0, 1, \dots, N-1\}$ 、 $0 \leq i \leq N-1$ である i が集合 $\{i_h, (i_h + 1) \bmod N\}$ の要素でないとして、

$k(i_h, h) = h + N + 1$ 、 $k(\bmod(i_h + 1) \bmod N, h) = k(i_h, h-2)$ 、及び $k(i, h) = k((i-1) \bmod N, h-1)$ 。

【0071】2) $0 \leq i \leq N-1$ である入力 i は、パケットを送信すべき出力 j を出力集合 $O_{k(i, h)}$ からRR方式で選択する。ただし、 $k(i, h) \neq 0$ である。なお、 $0 \leq i \leq N-1$ 及び $i \neq i_h$ である入力 i は先行するタイムスロットにおいて $(i-1) \bmod N$ の入力から集合 $O_{k(i, h)}$ を受信している。選択された出力 j は、受信した集合 $O_{k(i, h)}$ から除外される。

【0072】3) $0 \leq i \leq N-1$ である入力 i は、選択された出力 j のアドレスを接続メモリのメモリ位置 $k(i, h) \bmod (N+1)$ に記憶する。対応する受信入出力キューからライン先頭のHOLパケットが別個の送信入出力キューへ移される。

【0073】4) $0 \leq i \leq N-1$ で $i \neq (i_h - 2) \bmod N$ である入力 i （現アービタ）は、次の入力 $(i+1) \bmod N$ （後続するアービタ）へ集合 $O_{k(i, h)}$ を転送する。 $(i_h - 2) \bmod N$ の入力は集合 $O_{k((i_h - 2) \bmod N, h)}$ を1タイムスロット分だけ遅延させた後、転送する。

【0074】5) 続いて、クロスバースイッチ101は、 $0 \leq i \leq N-1$ である入力 i と、入力 i のメモリ位置 $(h \bmod (N+1))$ から読み込まれたアドレスの出力ポートとの間にクロスバークネクションを確立する。

【0075】6) $0 \leq i \leq N-1$ である各入力 i について、スケジュールされた送信入出力キューの先頭（HOL）のパケットをスイッチコアに確立された接続を通して選択された出力ラインへ送信する。

【0076】マルチキャスト・スケジューリング

本発明は更にマルチキャスト機能を有する。マルチキャスト・パケットは別個のキューに格納され到着順に処理される（FCFS: first come first served）。各キューは、そのHOLパケットのあて先を示すマルチキャストビット

マップ(BM)を有する。最も簡単なバージョンでは、マルチキャスト・キューはユニキャスト・キューよりも優先されて処理される。 h 番目のタイムスロットにおけるマルチキャストの場合、次の動作が付加される。

【0077】1) $0 \leq i \leq N-1$ である入力 i は、 $j \in O_{k(i, h)} \cap BM_i$ を満たす全ての出力 j を選択する。HOLマルチキャストパケットは、 k 番目のタイムスロットにおいて、選択された全ての出力ポートへ送信される。

【0078】2) $O_{k(i, h)} \cap BM_i$ が空集合であればユニキャストキューが動作する。それ以外の場合は、選択された出力が $O_{k(i, h)}$ 及び BM_i から除外される。

【0079】3) BM_i が空であれば、HOLマルチキャストパケットがマルチキャストキューから削除される。

【0080】スイッチコントローラの実装

図4は、 $N \times N$ 光クロスバースイッチの制御を行うアービタ部の構成を示すブロック図である。 N 個の入力ポート(0)～($N-1$)にそれぞれ対応して、入力モジュール $IM_0 \sim IM_{N-1}$ 、パイプライン処理を実行するRRアービタコントローラ $ARB_0 \sim ARB_{N-1}$ 、及びコネクションメモリ $M_0 \sim M_{N-1}$ が設けられている。RRアービタコントローラ $ARB_0 \sim ARB_{N-1}$ は、本発明によるパイプライン処理を行うためにリング上に接続されている。

【0081】各入力モジュール IM は、到着したパケットを論理的に分離したキュー（受信入出力キュー）に格納する。上述したように、各受信入出力キューによって、当該入力ポートがある特定の出力ポートへパケットを出力するように関連付けられている。入力モジュール IM は、到着パケットの出線リクエスト RQ によって各受信キューに格納した到着パケットの管理を行うと共に、出線リクエスト RQ を関連付けられたRRアービタコントローラ ARB へ送出する。

【0082】RRアービタコントローラ ARB は、入力した出線リクエスト RQ に応答して、利用可能出線情報 O から未来のタイムスロットで利用可能な出力ポートの1つを選択し、選択された出力ポート（出線番号） GR を関連付けられた入力モジュールへ戻す。上述したパイプラインの初期化プロセスによって、どのタイムスロットにおいても、RRアービタコントローラ ARB が同じタイムスロットに対して重複して送信スケジューリングを行わないことが保証される。

【0083】各入力モジュール IM は、更に、未来のタイムスロットにおいて出力ポートの予約（スケジューリング）に成功したパケットを別個の送信入出力キューとして格納する。また、RRアービタコントローラ ARB は、対応するコネクションメモリ M の特定場所にスケジューリング結果を書き込む。メモリのアドレスは、パケットがスケジューリングされたタイムスロットによって決定される。

【0084】RRアービタコントローラ ARB は、ある

タイムスロットで未だ予約されていない全ての出力ポートを示す制御情報を後続するRRアービタコントローラARBに通知する。より正確には、その制御情報によって予約済みの出力に対する出線リクエストが禁止される。もし制御情報を次のポートへ転送しないものがあるれば、そのRRアービタコントローラに続く次のRRアービタコントローラは未来のタイムスロットで任意の出力を選択できることとなる。

【0085】コネクションメモリに書き込まれたスケジュールに基づいて、パケットは入力モジュールからスイッチコア101を通して出力モジュール(OM)へ転送される。

【0086】なお、本発明の変更や変形は、上記記載及び教示から当業者には明らかであろう。ここでは本発明のいくつかの実施形態だけを記載したが、本発明の技術的範囲から逸脱することなく、多くの変形例をなすことは可能である。

【0087】性能比較

まず、RRGSと同程度の複雑さ(complexity)を有する他のプロトコルとの比較を行う。「複雑さ」は、1つのスケジュールを完了するのに必要な時間によって計られる。ここでは、HOL (Head Of Line)、I-TDMA (Interleaved TDMA)、SLIP (Iterative round-robin matching with slip)、RGS (Random greedy scheduling)、及びRRGSプロトコルの比較を行う。

【0088】HOLプロトコルは、入力キューを用いたスイッチのための最も簡単なプロトコルである(M. Karol et al., "Input vs. output queuing on a space-division packet switch," IEEE Transactions on Communications, vol. COM-35, no.12, Dec. 1987, pp. 1347-1356)。各入力には適切な出力へHOLパケットの送信要求を送出する。要求された出力は、ラウンドロビン方式で入力1つに対して送信許可を与える。次のタイムスロットで、送信許可を受けた入力パケットを対応する出力へ向かって送る。

【0089】インターリーブTDMA(I-TDMA)においては、出力は入力に固定化された方法により割り当てられる(K. Bogineni et al., "Low-complexity multiple access protocols for wavelength-division multiplexed photonic networks," IEEE Journal on Selected Areas in Communications, vol.11, no.4, May 1993, pp.590-604)。時間がフレームに分割され、そのフレームの各タイムスロットにおいて送信スケジュールが前もって決められている。パケットは、行く先に従って別個のキューに格納され、それらのスケジュールされたタイムスロットで送信される。

【0090】SLIPプロトコルは、N. McKeown et al. "Scheduling cells in an input-queued switch," Electronic Letters, vol.29, no.25, Dec. 1993, pp. 2174-2175に提案されている。各入力、送出すべきパ

ケットを有する場合、それらパケットの全ての送出先である出力に対して要求を出す。要求された出力は、ラウンドロビン方式で、要求を出している入力1つに対して送信許可を与える。複数の許可を受けた入力、ラウンドロビン方式で、許可された出力1つを選択する。ラウンドロビン選択は、前回選択された候補者の次から開始される。

【0091】RGSプロトコルは、ラウンドロビン選択をランダム選択に置き換えた点を除けば、RRGSと類似している(D. Guo et al., "Scalable high-speed protocols for WDM optical star networks," IEEE INFOCOM '94)。コントローラはランダムに一連の入力を選択し、それらを比較しなかった出力とランダムに比較する。高速になると、RGSは1タイムスロット内で作業を終了することが出来ない。しかしながら、本発明者等は、ランダム選択をラウンドロビン選択で置き換えた場合の影響を検査しその性能評価を行った。

【0092】図5は、RRGS、RGS、HOL、SLIP及びI-TDMAを実装した場合、提供されたトラフィック負荷に対する平均パケット遅延をそれぞれ示すグラフである。理論特性値とシミュレーション結果とはよく一致している。

【0093】図6(A)及び(B)は、RRGS、RGS、SLIP及びI-TDMAを実装した場合、固定負荷0.8及び0.9におけるパケット遅延の相補分布関数をそれぞれ示すグラフである。プロットされた曲線はシミュレーション結果に基づく。ほとんどの負荷に対して、RRGSはSLIP及びI-TDMAよりも性能がかなり優れている。

【0094】図7(A)は、RRGS、RGS、SLIP及びI-TDMAを実装しトラフィック負荷が非単調である条件下で、入出力キュー(i, j)のグループG1での平均パケット遅延をそれぞれ示すグラフであり、図7(B)は、同条件下で異なる入出力キューのグループG2での平均パケット遅延をそれぞれ示すグラフであり、図8(A)は、同条件下で異なる入出力キューのグループG3での平均パケット遅延をそれぞれ示すグラフであり、図8(B)は、同条件下で異なる入出力キューのグループG4での平均パケット遅延をそれぞれ示すグラフである。

【0095】

【発明の効果】本発明は、入力バッファ付高速スイッチのためのパイプライン・ラウンドロビン・スケジューリング方法を提供する。本発明によるRRGSプロトコルでは、図5～図8に示したように、同様の複雑さにおいて、他のプロトコルよりもパケットの平均遅延時間が短い。また、パケット遅延分布が大きく広がっていない。トラフィック負荷が非単調である場合、他のプロトコルに比べると、負荷の少ないキューの遅延時間は長くなるが、負荷の大きいキューの遅延時間は遥かに短くなっている。

【図面の簡単な説明】

【図1】本発明による超高速交換システムの概念的アー

キテクチャを示すブロック図である。

【図2】奇数入力数の 5×5 クロスバースイッチを用いた場合のRRGSスケジューリングの一例を示すタイミングチャートである。

【図3】偶数入力数の 4×4 クロスバースイッチを用いた場合のRRGSスケジューリングの一例を示すタイミングチャートである。

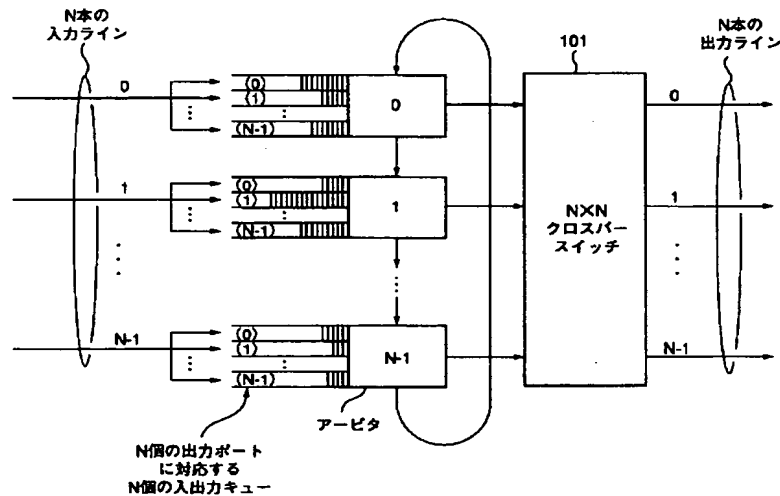
【図4】 $N \times N$ 光クロスバースイッチの制御を行うアービタ部の構成を示すブロック図である。

【図5】RRGS, RGS, HOL, SLIP及びI-TDMAを実装した場合、提供されたトラフィック負荷に対する平均バケット遅延のシミュレーション結果と理論値とをそれぞれ示すグラフである。

【図6】(A)及び(B)は、RRGS, RGS, SLIP及びI-TDMAを実装した場合、固定負荷0.8及び0.9におけるバケット遅延の相補分布関数をそれぞれ示すグラフである。

*

【図1】



*【図7】(A)は、RRGS, RGS, SLIP及びI-TDMAを実装しトラフィック負荷が非単調である条件下で、入出力キュー(i, j)のグループG1での平均バケット遅延をそれぞれ示すグラフであり、(B)は、同条件下で異なる入出力キューのグループG2での平均バケット遅延をそれぞれ示すグラフである。

【図8】(A)は、同条件下で異なる入出力キューのグループG3での平均バケット遅延をそれぞれ示すグラフであり、(B)は、同条件下で異なる入出力キューのグループG4での平均バケット遅延をそれぞれ示すグラフである。

【符号の説明】

101 $N \times N$ クロスバースイッチ

IM 入力モジュール

ARB RRアービタ及びパイプライン・コントローラ

M コネクションメモリ

【圖3】

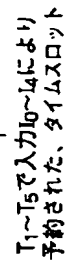
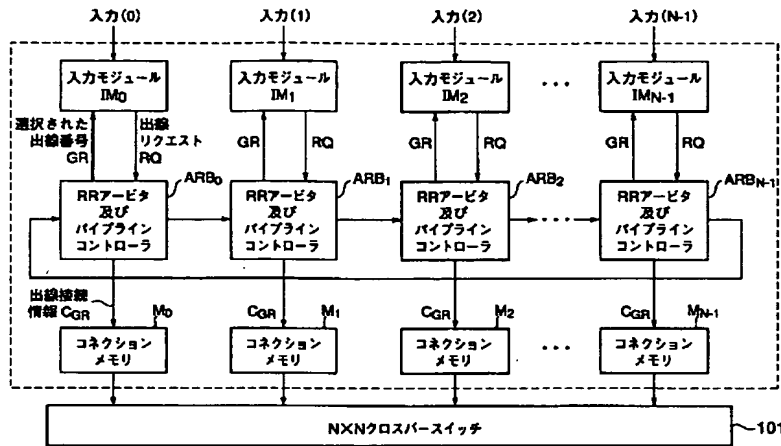
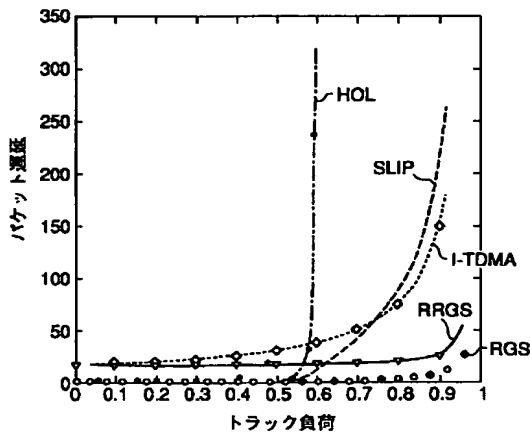


Figure 1 is a Gantt chart illustrating the execution of three tasks (1, 2, 3) over time. The tasks are represented by horizontal bars. Task 1 (1) has a duration of 10 units. Task 2 (2) has a duration of 10 units. Task 3 (3) has a duration of 10 units. The chart is divided into three regions: 301, 302, and 303. Region 301 contains the first 10 units of Task 1. Region 302 contains the next 10 units of Task 1 and the first 10 units of Task 2. Region 303 contains the next 10 units of Task 1, the next 10 units of Task 2, and the first 10 units of Task 3. The tasks are executed in parallel, with Task 1 starting at time 0 and Task 2 starting at time 10. Task 3 starts at time 20. The chart shows that Task 1 is completed at time 10, Task 2 is completed at time 20, and Task 3 is completed at time 30.

【図4】

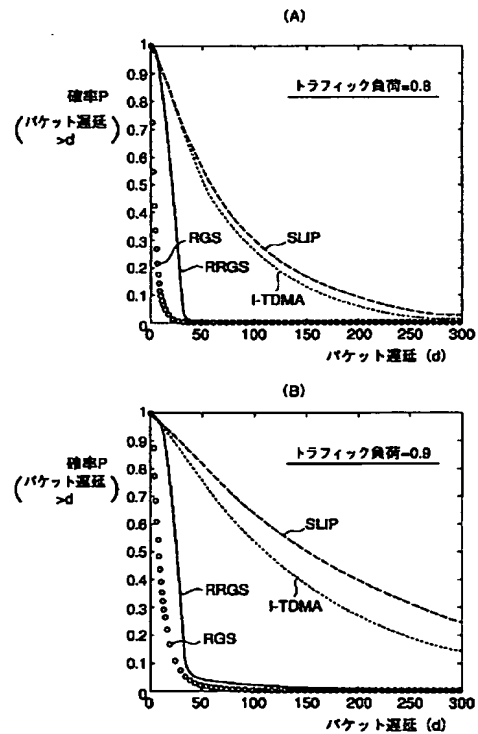


【図5】

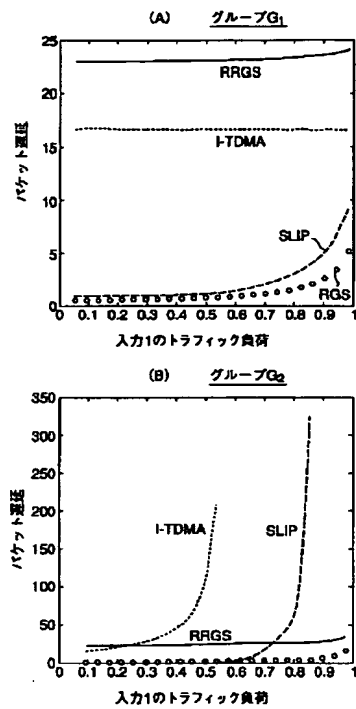


RRGS { シミュレーション: —
理論値: ▽
RGS { シミュレーション: ○
理論値: +
HOL { シミュレーション: - - -
理論値: ●
SLIP { シミュレーション: ·····
理論値: ◇
I-TDMA { シミュレーション: - · - · -
理論値: ◇

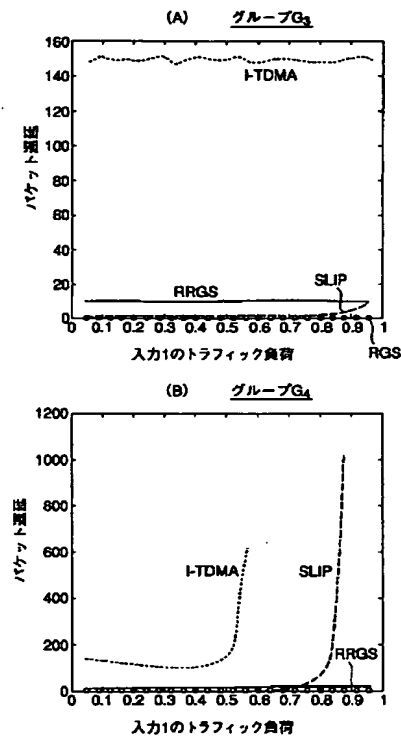
【図6】



【図7】



【図8】



フロントページの続き

(72)発明者 ルイクシュー ファン
アメリカ合衆国、ニュージャージー、
08540 プリンストン、インディペンデンス
ウェイ 4、エヌ・イー・シー・ユー・
エス・エー・インク内

(72)発明者 アレクサンドラ スミルジャンク
アメリカ合衆国、ニュージャージー、
08540 プリンストン、インディペンデンス
ウェイ 4、エヌ・イー・シー・ユー・
エス・エー・インク内